



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/679,008	10/02/2003	Robert C. Chang	SANDP036	8904

10027 7590 08/09/2007
ANDERSON, LEVINE & LINTEL L.L.P.
14785 PRESTON ROAD
SUITE 650
DALLAS, TX 75254

EXAMINER

TSAI, SHENG JEN

ART UNIT	PAPER NUMBER
----------	--------------

2186

MAIL DATE	DELIVERY MODE
-----------	---------------

08/09/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/679,008

Applicant(s)

CHANG ET AL.

Examiner

Sheng-Jen Tsai

Art Unit

2186

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 20 June 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1, 3-8, 10, 15-16, 19- 22 and 43-48 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1, 3-8, 10, 15-16, 19- 22 and 43-48 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 02 October 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) ☒ Notice of References Cited (PTO-892) | 4) ☐ Interview Summary (PTO-413) |
| 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) ☐ Information Disclosure Statement(s) (PTO/SB/08) | 5) ☐ Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) ☐ Other: _____ |

DETAILED ACTION

1. This Office Action is taken in response to Applicant's Remarks filed on June 20, 2007 regarding application 10,679,008 filed on October 2, 2003.
2. Claims 1, 3-8, 10, 15-16, 19-22 and 43-48 are pending for consideration.

3. ***Response to Remarks***

Applicants' remarks have been fully and carefully considered.

In view of these remarks, rejections of claims 1, 3-6, 8-10, 15-16, 19-20, 22 and 43-48 under 35 U.S.C. 102(b) as being anticipated by Achiwa et al. (US 5,737,742) have been withdrawn.

In response to the remarks, a new ground of claim analysis based on a newly identified reference (Miyauchi, US 6,430,650) has been made. Refer to the corresponding sections of the following claim analysis for details.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1, 3-6, 8-10, 15-16, 19-20, 22 and 43-48 are rejected under 35 U.S.C. 102(e) as being anticipated by Miyauchi (US 6,430,650).

As to claim 1, Miyauchi discloses **a method** [Semiconductor Storage Device for Group management of Data Sectors (title); abstract] **for performing a write operation** [figures 7A and 7B show the flowchart of write operation] **in a non-volatile memory system** [The corresponding non-volatile memory system is the flash memory system shown in figure 1, 11, which includes a flash memory (figure 1, 15) which is non-volatile], **the non-volatile memory system including a non-volatile memory** [The corresponding non-volatile memory system is the flash memory system shown in figure 1, 11, which includes a flash memory (figure 1, 15) which is non-volatile] **having a plurality of erase units** [FIG. 2 shows an erase block 20 in a flash memory 15. A block size is 64 kilobytes. ... (column 3, lines 66-67)); the total capacity of the storage device is more than 1 megabytes (column 1, lines 54-60; column 3, lines 42-50), thus it has a plurality of erase units, each of them with a size of 64 kbytes], **each erase unit including a plurality of programming units arranged in a sequence** [The corresponding programming units are the sectors: "Though read and write can be performed in the unit of a sector (512 bytes), erase can only be performed in the unit of a relatively large size such as 64 kbytes (column 1, lines 33-35)"; since each erase unit is 64 kbytes in size and each sector has 512 bytes, an erase unit includes a plurality of 128 sectors (i.e., the programming units); figure 3 shows that sectors 0, 1, 2 and 3 of PGN=X are arranged in a sequence (column 3, lines 26-37)], **each programming unit comprising a plurality of non-volatile memory cells** [a sector has 512 bytes (column 1, lines 33-35), note that each "bit" of the 512 bytes represents a memory cell] **wherein a logical unit is associated with a first erase unit and includes a plurality**

of logic sub-units arranged in a sequence [figures 4 and 6 show the correspondence between logical entities and physical entities; column 3, lines 30-37; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7), note that each "logical sector number" represents a "logic sub-unit," and the collection of all logical sector numbers associated with a block represents a "logical unit"], **the method comprising:**

grouping the programming units [The corresponding programming units are the sectors (Though read and write can be performed in the unit of a sector (512 bytes) (column 1, lines 33-35))] **within the first erase unit into a plurality of groups of programming units, the groups arranged in a sequence** [A semiconductor storage device comprises a flash memory and manages the sectors in the unit of groups each consisting of n sectors (abstract); The semiconductor storage device 11 manages four sectors as one group (column 3, lines 26-27); figure 5 shows a plurality of groups in sequence].

writing contents of a first logical sub-unit into a first programming unit in a first group of programming units of the first erase unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 4 into

Art Unit: 2186

the flash memory (see figures 7A~7B, note that 4 divided by 4 gives a quotient of 1 and a remainder of 0), and if sector 0 of group PGN=1 has not been written with data yet, then the contents of logical sector number 4 (i.e., the first logical sub-unit) from the host will be written into sector 0 (i.e., the first programming unit) of group PGN=1 (i.e., the first group) because this is a "new write" (column 4, line 58 to column 5, line 7; column 5, line 63 to column 6, line 27); figures 5 and 8A~8E]; **and**

then writing contents of a second logical sub-unit into a second programming unit in a first group of programming units [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 6 into the flash memory (see figures 7A~7B, note that 6 divided by 4 gives a quotient of 1 and a remainder of 2), and if sector 2 of group PGN=1 has not been written with data yet, then the contents of logical sector number 6 (i.e., the second logical sub-unit) from the host will be written into sector 2 (i.e., the second programming unit) of group PGN=1 (i.e., the first group) because this is a "new write" (column 4, line 58 to column 5, line 7; column 5, line 63 to column 6, line 27); figures 5 and 8A~8E], **the second logical sub-unit later in the sequence of logical sub-units than the first logical sub-unit, and the second programming unit later in the sequence of the programming units**

than the first programming unit [logical sector number 6 (i.e., the second logical sub-unit) is later than logical sector number 4 (i.e., the first logical sub-unit)],
then writing contents of a third logical sub-unit of the first logic unit into a programming unit in a second group of programming units of the first erase unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set as "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 5 into the flash memory (see figures 7A~7B, note that 5 divided by 4 gives a quotient of 1 and a remainder of 1), and if sector 1 of group PGN=1 has already been written with data, then the contents of logical sector number 5 (i.e., the third logical sub-unit) from the host will be written into sector 1 of group PGN=2 (i.e., the second group) because this is a "additional write" (column 4, line 58 to column 5, line 28; column 6, lines 29-61); figures 5 and 8A~8E], **the third logical sub-unit earlier in the sequence of logical sub-units than the second logical sub-unit** [logical sector number 5 (the third logical sub-unit) is earlier than logical sector number 6 (i.e., the second logical sub-unit)].

As to claim 3, Miyauchi teaches that **the method of claim 1, further comprising:**

After the step of writing contents of a third logic sub-unit, writing contents of a fourth logical sub-unit into a programming unit of the second group, the fourth

logical sub-unit later in the sequence of logical sub-units than the third logical sub-unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 7 into the flash memory (see figures 7A~7B, note that 7 divided by 4 gives a quotient of 1 and a remainder of 3), and if sector 3 of group PGN=1 has already been written with data, then logical sector number 7 (i.e., the fourth logical sub-unit) from the host will be written into sector 3 of group PGN=2 (i.e., the second group) because this is a "additional write" (column 4, line 58 to column 5, line 28; column 6, lines 29-61); logical unit number 7 (i.e., the fourth logical sub-unit) is later than logical unit number 5 (i.e., the third logical sub-unit); figures 5 and 8A~8E].

As to claim 4, Miyauchi teaches that **the method of claim 1, further comprising:**

writing a first identifier into the first group of programming units [figure 5 shows that the first group has an identifier PGN (Physical Group Number)=X and the second group has an identifier PGN=Y; figures 8A~8E and 9A~9E show more examples; the Logical Group Number is stored in the group management information (figure 2, 22; column 4, lines 1-25)], **the first identifier being arranged to substantially identify the first logical sub-unit** [figures 5 and 6 show the correspondence between the identifiers and the logical sub-units; a logical sector number is converted to a logical

group number and an offset address, and a conversion table is provided to convert a logical group number to a physical group number (abstract)];

writing contents of a first logical sub-unit of a second logical unit into a programming unit of a second group of programming units in the first erase unit

[figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 8 into the flash memory (see figures 7A~7B, note that 8 divided by 4 gives a quotient of 2 and a remainder of 0), and if sector 0 of group PGN=2 has not been written with data yet, then the contents of logical sector number 8 (i.e., the first logical sub-unit) from the host will be written into sector 0 (i.e., the first programming unit) of group PGN=2 (i.e., the second group) because this is a "new write" (column 4, line 58 to column 5, line 7; column 5, line 63 to column 6, line 27); figures 5 and 8A~8E]; **and**

writing a second identifier into the second group of programming units [figure 5 shows that the first group has an identifier PGN (Physical Group Number)=X and the second group has an identifier PGN=Y; figures 8A~8E and 9A~9E show more examples; the Logical Group Number is stored in the group management information (figure 2, 22; column 4, lines 1-25)], **the second identifier being arranged to substantially identify the second logical sub-unit** [figures 5 and 6 show the

correspondence between the identifiers and the logical sub-units; a logical sector number is converted to a logical group number and an offset address, and a conversion table is provided to convert a logical group number to a physical group number (abstract)].

As to claim 5, Miyauchi teaches that **each of the erase units is a physical block, and the first logic unit is a logic block** [figures 5 and 6 show the correspondence between the physical entities and the logical sub-units; column 3, line 66 to column 4, line 40], **wherein each of the programming units is a physical page** [the sectors (column 3, lines 42-52)]; **and wherein each of the logical sub-units is a logical page** [the logical sector number from the host (abstract; column 6, lines 4-7)].

As to claim 6, Miyauchi teaches that **the non-volatile memory is a flash memory** [flash memory, figure 1, 15].

As to claim 8, Miyauchi teaches that **the method of claim 1, wherein the contents of the third logical sub-unit substantially supercede the contents of the first logical sub-unit** [this corresponds to the “additional write” (column 4, line 58 to column 5, line 28) wherein the third logical unit number and the first logical unit number from the host are the same].

As to claim 10, the limitations recited in this claim are substantially the same as those recited in claims 1 and 4. Thus claim 10 is rejected by the same reasons as applied to claim 1 and claim 4. Refer to “As to claim 1” and “As to claim 4” presented earlier in this section for details.

As to claim 15, Miyauchi teaches that **the non-volatile memory system** [figure 18 shows the memory system], **wherein the controlling means comprises: code devices for performing the sequence of operations** [figures 7A~7B and figure 10 show the code that control the operation of the storage device; the code device comprises memory that stores the code and the address conversion table (figure 1, 16)]; **and a processor for accessing and executing the code devices** [CPU (figure 1, 13) is the processor; the CPU controls the operation of the entire semiconductor storage device (column 3, lines 24-25)].

As to claim 16, Miyauchi teaches that **the method of claim 22, wherein the contents of the third logical sub-unit substantially supercede the contents of the first logical sub-unit** [this corresponds to the "additional write" (column 4, line 58 to column 5, line 28) wherein the third logical unit number and the first logical unit number from the host are the same].

As to claim 19, Miyauchi teaches that **each of the erase units is a physical block, and the first logic unit is a logic block** [figures 5 and 6 show the correspondence between the physical entities and the logical sub-units; column 3, line 66 to column 4, line 40], **wherein each of the programming units is a physical page** [the sectors (column 3, lines 42-52)]; **and wherein each of the logical sub-units is a logical page** [the logical sector number from the host (abstract; column 6, lines 4-7)].

As to claim 20, Miyauchi teaches that **the non-volatile memory is a flash memory** [flash memory, figure 1, 15].

As to claim 22, Miyauchi discloses **a non-volatile memory system** [The corresponding non-volatile memory system is the flash memory system shown in figure 1, 11, which includes a flash memory (figure 1, 15) which is non-volatile], **comprising: a non-volatile memory** [The corresponding non-volatile memory system is the flash memory system shown in figure 1, 11, which includes a flash memory (figure 1, 15) which is non-volatile] **having a plurality of erase units** [FIG. 2 shows an erase block 20 in a flash memory 15. A block size is 64 kilobytes. ... (column 3, lines 66-67)); the total capacity of the storage device is more than 1 megabytes (column 1, lines 54-60; column 3, lines 42-50), thus it has a plurality of erase units, each of them with a size of 64 kbytes], **each erase unit including a plurality of programming units arranged in a sequence** [The corresponding programming units are the sectors: "Though read and write can be performed in the unit of a sector (512 bytes), erase can only be performed in the unit of a relatively large size such as 64 kbytes (column 1, lines 33-35)"; since each erase unit is 64 kbytes in size and each sector has 512 bytes, an erase unit includes a plurality of 128 sectors (i.e., the programming units); figure 3 shows that sectors 0, 1, 2 and 3 of PGN=X are arranged in a sequence (column 3, lines 26-37)], **each programming unit comprising a plurality of non-volatile memory cells** [a sector has 512 bytes (column 1, lines 33-35), note that each "bit" of the 512 bytes represents a memory cell] **wherein a logical unit is associated with a first erase unit and includes a plurality of logic sub-units arranged in a sequence** [figures 4 and 6 show the correspondence between logical entities and physical entities; column 3, lines 30-37; For example, when the logical sector numbers designated by the host

10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7), note that each "logical sector number" represents a "logic sub-unit," and the collection of all logical sector numbers associated with a block represents a "logical unit";

means for controlling the writing of contents into the non-volatile memory [figure 1 shows the means, including the CPU (13) and flash controller (14); column 3, lines 16-25); figures 7A~7B shows the flowchart of writing the contents into the flash memory] **according to groups of programming units within the erase units, the groups arranged in sequence** [A semiconductor storage device comprises a flash memory and manages the sectors in the unit of groups each consisting of n sectors (abstract); The semiconductor storage device 11 manages four sectors as one group (column 3, lines 26-27); figure 5 shows a plurality of groups in sequence], **according to a sequence of operations comprising:**

writing contents of a first logical sub-unit into a first programming unit in a first group of programming units of the first erase unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 4 into the flash memory (see figures 7A~7B, note that 4 divided by 4 gives a quotient of 1 and a remainder of 0), and if sector 0 of group PGN=1 has not been written with data yet,

Art Unit: 2186

then the contents of logical sector number 4 (i.e., the first logical sub-unit) from the host will be written into sector 0 (i.e., the first programming unit) of group PGN=1 (i.e., the first group) because this is a "new write" (column 4, line 58 to column 5, line 7; column 5, line 63 to column 6, line 27); figures 5 and 8A~8E]; **and**

then writing contents of a second logical sub-unit into a second programming unit in a first group of programming units [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 6 into the flash memory (see figures 7A~7B, note that 6 divided by 4 gives a quotient of 1 and a remainder of 2), and if sector 2 of group PGN=1 has not been written with data yet, then the contents of logical sector number 6 (i.e., the second logical sub-unit) from the host will be written into sector 2 (i.e., the second programming unit) of group PGN=1 (i.e., the first group) because this is a "new write" (column 4, line 58 to column 5, line 7; column 5, line 63 to column 6, line 27); figures 5 and 8A~8E], **the second logical sub-unit later in the sequence of logical sub-units than the first logical sub-unit, and the second programming unit later in the sequence of the programming units than the first programming unit** [logical sector number 6 (i.e., the second logical sub-unit) is later than logical sector number 4 (i.e., the first logical sub-unit)],

then writing contents of a third logical sub-unit of the first logic unit into a programming unit in a second group of programming units of the first erase unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 5 into the flash memory (see figures 7A~7B, note that 5 divided by 4 gives a quotient of 1 and a remainder of 1), and if sector 1 of group PGN=1 has already been written with data, then the contents of logical sector number 5 (i.e., the third logical sub-unit) from the host will be written into sector 1 of group PGN=2 (i.e., the second group) because this is a "additional write" (column 4, line 58 to column 5, line 28; column 6, lines 29-61); figures 5 and 8A~8E], **the third logical sub-unit earlier in the sequence of logical sub-units than the second logical sub-unit** [logical sector number 5 (the third logical sub-unit) is earlier than logical sector number 6 (i.e., the second logical sub-unit)].

As to claim 43, Miyauchi teaches that **the method of claim 1, further comprising:**

after the step of writing contents of the third logical sub-unit, receiving contents of a fourth logical sub-unit to be written into the non-volatile memory, the fourth logical sub-unit earlier in the sequence of logical sub-units than the third logical sub-unit [figures 7A and 7B show the flowchart of write operation on how the content

of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 7 into the flash memory (see figures 7A~7B, note that 7 divided by 4 gives a quotient of 1 and a remainder of 3), and if sector 3 of group PGN=1 has already been written with data, then logical sector number 7 (i.e., the fourth logical sub-unit) from the host will be written into sector 3 of group PGN=2 (i.e., the second group) because this is a "additional write" (column 4, line 58 to column 5, line 28; column 6, lines 29-61); logical unit number 7 (i.e., the fourth logical sub-unit) is later than logical unit number 5 (i.e., the third logical sub-unit); figures 5 and 8A~8E]; **determining whether one of the groups of programming units in the first erase unit is available for writing into** [figure 7A, step S103; column 5, line 63 to column 6, line 28]; **responsive to none of the groups of programming units being available for writing into** [if the answer is "YES" for step S117 of figure 7B], **merging the contents of the first logical unit that were previously written into the plurality of programming units of the first erase unit into programming units of a first group of programming units in a second erase unit** [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36]; and

then writing the contents of the fourth logical sub-unit into a programming unit of a second group of programming units of the second erase unit [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written (i.e., existing) data; column 7, lines 20-36].

As to claim 44, Miyauchi teaches that **the method of claim 43, wherein the merging step merges the contents of the first logical unit that were previously written into the plurality of programming units of the first erase unit with contents of the first logical unit stored in another erase unit** [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written (i.e., existing) data; column 7, lines 20-36].

As to claim 45, Miyauchi teaches that **the method of claim 43, further comprising:**
determining whether additional contents of the first logical unit remain to be updated [figures 7A~7B; column 4, line 58 to column 5, line 28];
responsive to additional contents remaining to be updated, repeating the determining, merging, and writing steps [figure 7A, step S103; column 5, line 63 to column 6, line 28; figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36]; **and**

responsive to additional contents not remaining to be updated, merging the contents associated with the first logical unit and stored in the second erase unit into a third erase unit [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36].

As to claim 46, Miyauchi teaches that **the system of claim 23, further comprising:**

after the step of writing contents of the third logical sub-unit, receiving contents of a fourth logical sub-unit to be written into the non-volatile memory, the fourth logical sub-unit earlier in the sequence of logical sub-units than the third logical sub-unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 7 into the flash memory (see figures 7A~7B, note that 7 divided by 4 gives a quotient of 1 and a remainder of 3), and if sector 3 of group PGN=1 has already been written with data, then logical sector number 7 (i.e., the fourth logical sub-unit) from the host will be written into sector 3 of group PGN=2 (i.e., the second group) because this is a "additional write" (column 4, line 58 to column 5,

line 28; column 6, lines 29-61); logical unit number 7 (i.e., the fourth logical sub-unit) is later than logical unit number 5 (i.e., the third logical sub-unit); figures 5 and 8A~8E]; **determining whether one of the groups of programming units in the first erase unit is available for writing into** [figure 7A, step S103; column 5, line 63 to column 6, line 28];

responsive to none of the groups of programming units being available for writing into [if the answer is "YES" for step S117 of figure 7B], **merging the contents of the first logical unit that were previously written into the plurality of programming units of the first erase unit into programming units of a first group of programming units in a second erase unit** [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36]; **and**

then writing the contents of the fourth logical sub-unit into a programming unit of a second group of programming units of the second erase unit [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written (i.e., existing) data; column 7, lines 20-36].

As to claim 47, Miyauchi teaches that **the system of claim 46, wherein the merging step merges the contents of the first logical unit that were previously written into the plurality of programming units of the first erase unit with contents of the first logical unit stored in another erase unit** [figure 7B, steps S120

to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written (i.e., existing) data; column 7, lines 20-36].

As to claim 48, Miyauchi teaches that **the system of claim 46, further comprising:**

determining whether additional contents of the first logical unit remain to be updated [figures 7A~7B; column 4, line 58 to column 5, line 28];

responsive to additional contents remaining to be updated, repeating the determining, merging, and writing steps [figure 7A, step S103; column 5, line 63 to column 6, line 28; figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36]; **and**

responsive to additional contents not remaining to be updated, merging the contents associated with the first logical unit and stored in the second erase unit into a third erase unit [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36].

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 7 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Miyauchi (US 6,430,650), and in view of Kramer (US 6,182,239).

As to claims 7 and 21, Miyauchi does not mention that **the non-volatile memory is one of a NAND flash memory and an MLC NAND flash memory.**

However, the invention of Miyauchi is directly applicable to any type of flash memories, including NAND flash memory and MLC NAND flash memory.

Further, Kramer teaches in the invention "Fault-Tolerant Codes for Multi-Level Memories" a fault-tolerant code semiconductor flash memory storage devices including an array of individual multi-level cell (MLC) storage devices [abstract; column 2, lines 1-15] as well as NAND flash memory [column 2, lines 36-57].

Therefore, it would have been obvious for one of ordinary skills in the art at the time of Applicants' invention to adopt NAND type flash memory as well as MLC NAND type of flash memory in a non-volatile memory system, because Kramer teaches that they offer special advantages [A flash memory with multilevel cell significantly reduces the memory per-bit cost. For instance, a 32 Mb multilevel cell (MLC) Flash memory storing two bits of data per cell achieves 32 Mb memory storage capacity using 16M Flash memory cells (column 2, lines 5-10); a 128 Mb multi-level NAND flash memory stores two bits per cell by tight programmed cell threshold voltage (V_{th}) control. Program states have 0.4V V_{th} distribution and a 0.8V separation gap. Incremental-

8. *Related Prior Art On Record*

- Marshall et al., (US 5,832,493), "Flash File management System."
- Tomori et al., (US patent Application Publication 2002/0085433), "Data management System and Data management Method."
- Achiwa et al., (US 5,737,742), "memory system Using Flash memory and Method of Controlling the memory System."
- Sukegawa, (US 5,572,466), "Flash Memory Chips."
- Hong et al., (US 6,236,593), "Method for Simultaneously Programming Plural Flash Memories Having Invalid Blocks."
- Hewitt et al., (US 5,488,711), "Serial EEPROM device and Associated Method for Reducing Data Load Time Using a Page Mode Write Cache."
- Lofgren et al., (US 6,230,233), "Wear Leveling Techniques for Flash EEPROM Systems."
- Jou et al., (US 5,568,423), "Flash Memory Wear Leveling System Providing Immediate Direct Access to Microprocessor."
- Miyauchi, (US 6,970,890), "Method and Apparatus for Data Recovery."
- Sukegawa et al., (US 5,603,001), "semiconductor Disk System Having a Plurality of Flash Memories."

- Norman, (US 5,754,567), "Write Reduction in Flash Memory systems through ECC Usage."
- Bruce et al., (US 6,000,006), "Unified Re-Map and Cache-Index Table with Dual Write-Counters for wear-Leveling of Non-Volatile Flash RAM Mass Storage."

Conclusion

9. Claims 1, 3-8, 10, 15-16, 19- 22 and 43-48 are rejected as explained above.
10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sheng-Jen Tsai whose telephone number is 571-272-4244. The examiner can normally be reached on 8:30 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Sheng-Jen Tsai
Examiner
Art Unit 2186

August 2, 2007

Sheng Jen Tsai
8/3/07